Assessed Lab 1 (20%)

Due: 10:50 on 23 Oct 2017

Name:	
User ID:	

1 Rules

- **Do not turn over the page until asked to do so.**
- **This sheet must not leave the lab.**
- **You must work on your own.** Chat, email, message programs, and mobile phones must not be used during the assessed lab. If you talk to other students, or communicate by other means, you will receive a mark of 0.
- **You may not download Java code from any website except the module web page.** You can refer to code that you have written in previous lab classes. You may use any resources from the module web site (lecture slides, example code), from the Java API documentation, and you may refer to textbooks and lecture notes. You may use Google Translate if English is not your native language. **You may not refer to any other resources.**

Module website:	`http://www.dcs.shef.ac.uk/~adam/stuff/COM6516/`
Java API:	`https://docs.oracle.com/javase/8/docs/api/`

- Upload your `*.java` files to MOLE using the assignments button on the menu bar. Do not submit any other files, such as `*.class` files. (Ignore the earlier handout about using ZipCentral.)

  Your code must compile and run under Java 1.8, using `javac` and `java` on the command line. You can use the `sheffield` package already provided in this module and you may assume that it will be available to compile and run your code. You do not need to submit any of the `sheffield` files through MOLE.
- When you are finished, **you must hand in this sheet** with your name and CiCS user ID above. If you are satisfied with your work and submit before the deadline, you may leave the lab early without talking to any other students.
- You must stop working at the end of the lab session and submit your program code as described above. You should submit your code by 10:50; after this, MOLE will record the work as late and there will be a 10% penalty when it is marked. You will not be able to submit any code after 11:00.

  You can submit your work more than once; the last version submitted will be marked.
- If you are unable to upload your code to MOLE, please email your `*.java` files as attachments to `a.funk@sheffield.ac.uk` with the subject line "COM6516: Assessed Lab 1".
- You may ask the demonstrators questions only if you are having technical problems with the lab machines or you do not understand what is being asked for in the assignment. You may not ask them questions about how to use particular Java constructs.

2 Task

Read the following instructions carefully.

The programming task is to implement a simple arithmetic game for children.

Your program should start by asking for the name of the child. She types in her name and is then given 10 random arithmetic questions, involving addition and subtraction (hint: `java.util.Random`'s `nextBoolean()`, where the operands and answers are all in the range 0..20 (hint: `random.nextInt(i)` returns $0 \leq x < i$). The child should be told what the correct answer is if she gets the answer wrong.

After 10 questions have been answered the player's score is displayed, along with a relevant message, for example, "Well done". There should be 11 different messages, one for each score from 0 to 10, and the child's name should be included in the message. An example output is given here:

```
$ java Game
Please type in your name: Victoria
Hello Victoria

Question 1: 6 - 5
Answer? 1
Correct, well done
...
Question 9: 17 - 9
Answer? 4
The correct answer is 8

Question 10: 17 + 1
Answer? 4
The correct answer is 18

You scored 4 out of 10
A good effort Victoria
```

You need to write a class called `Game` (containing a `main` method), and a class called `Question` that generates a single question and answer in its constructor. The generated question and answer could be stored as instance fields. You must decide what are the useful attributes and methods of each class. A tricky part is designing the algorithm to generate a random question, since all operands *and answers* must be in the range 0..20.

Upload your `Game.java` and `Question.java` files through MOLE. Your solution must compile and run under Java 1.8 on the command line using `javac Game.java Question.java` and run using `java Game`.

Remember that you need to create an instance of `Random`; the methods in the hints above are not static. You do not need to worry about seeding the random number generator or handling invalid input (assume the user will only enter a name and then valid integers).

3 Marking

Your submission will be marked out of 20, with up to 10 marks for code that compiles and produces correct output, and 10 marks for good programming style, including consistent indentation, sensible use of comments, and appropriate use of Java programming constructs.

Marks and feedback will be returned to you through MOLE within 3 weeks.